**Department of Computer Applications**

**Gist of Lectures as per Syllabus of First Semester**

**Session 2017 – 2018**

 **Paper – KU BCA 103 PROGRAMMING IN C**

**Mohammad Asghar Ali**

**Assistant Professor**

**Dept of Computer Applications**

**Topic 1: Introduction of C programming**

C is a third generation procedural programming language. It was developed in the early 1970s by Dennis Ritchie at Bell Laboratories to be used by the UNIX operating system.

There are many importance like it is robust language, built-in function, portable etc.

A C program basically consists of the following parts:

1) Preprocessor Commands 2) Functions 3) Variables4) Statements & Expressions 5) Comments

Let us look at a simple code that would print the words "Hello World":

#include <stdio.h>

int main()

{

/\* my first program in C \*/

printf ("Hello, World! \n");

return 0;

}

Let us take a look at the various parts of the above program:

1. The first line of the program *#includes <stdio.h>* is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation.

2. The next line *int main ()* is the main function where the program execution begins.

3. The next line /\*...\*/ will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.

4. The next line *printf(...)* is another function available in C which causes the message "Hello, World!" to be displayed on the screen.

5. The next line return 0**;** terminates the main() function and returns the value 0.

**Topic 2: Tokens**

Token is a smallest unit of prg. A C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol .Identifier means the name of the variable,function,array etc.

Keyword means reserved word of c prg.

Data types means which type of data we are using .it has three parts primary data types, user defined data, derived data types etc.

Constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals.

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators:

1) Arithmetic Operators 2) Relational Operators 3)Logical Operators 4)Bitwise Operators 5) Assignment Operators 6) Increment and Decrement Operators 7)Conditional Operators 8) Special Operators

**Topic 3: Input and Output operations**

We can input data in formatted or unformatted form.

Formatted form-scanf (), printf ().

Unformatted form – getchar (), putchar (), gets (), puts ().

**Topic 4: Decision Making and Branching Decision**

Decision-making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

C language has different types of selection statements like

If statement, switch statement, condition operator,goto statement etc.

**Topic 5: Loop**

A loop statement allows us to execute a statement or group of statements multiple times

There are two types of loop one is finite and second is infinite loop.

Infinite means which has no end. Finite means which has end

Finite loop is divided into three parts while, do-while, for loop etc.

**Topic 6: Array**

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type. All arrays consist of contiguous memory locations.

There are different types of array like one dimensional Arrays Two-dimensional Arrays Multidimensional Arrays, Dynamic Arrays etc

**Topic 7: String**

String is a sequence of characters that is treated as a single data item. Any group of characters defined between double quotation marks (“”) is a string constant. Strings are actually one-dimensional array of characters terminated by a null character '\0'. There are also different types of built in function for string manipulation like strlen (),strrev() etc.

**Topic 8: Function**

A function is a group of statements that together perform a task. Every C program has at least one function, which is main (), and all the most trivial programs can define additional functions.

A function declarationtells the compiler about a function's name, return type, and parameters. A function definitionprovides the actual body of the function.

The C standard library provides numerous built-in functions that your program can call. For example, strcat()to concatenate two strings, memcpy()to copy one memory location to another location, and many more functions.

A function can also be referred as a method or a sub-routine or a procedure, etc.

**Topic: 9 Structures**

Arrays allow defining type of variables that can hold several data items of the same kind. Similarly, structureis another user-defined data type available in C that allows to combine data items of different kinds.

To define a structure, you must use the structstatement. The struct statement defines a new data type, with more than one member.

To access any member of a structure, we use the member access operator (.).

A unionis a special data type available in C that allows to store different data types in the same memory location.

Bit Fields allow the packing of data in a structure. This is especially useful when memory or data storage is at a premium

**Topic 10: Pointers**

A pointeris a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is:

type \*var-name;

there are two types of operators used in pointer one is ampersand(&) which is used to find the address of the variable and second is unary operator( **\*)** that returns the value of the variable located at the address specified by its operand.

A pointer to a pointer is a form of multiple indirection, or a chain of pointers.

**Topic 11: File Management in C.**

A file represents a sequence of bytes, regardless of it being a text file or a binary file. C programming language provides access on high-level functions as well as low-level (OS level) calls to handle file on your storage devices. You can use different types of function for file handling.

You can use the fopen ( )function to create a new file or to open an existing file. he prototype of this function call is as follows:

FILE \*fopen (const char \* filename, const char \* mode );

To close a file, use the fclose ( ) function.

To reads a character from a file, use the getc ().

To writes a character to file, use the putc ().

There are many more function for file handling

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Third Semester**

**Session 2017 – 2018**

**Paper – KU BCA 303 RELATIONAL DATABASE MANAGEMENT SYSTEMS**

**Topic 1: DATABASE SYSTEM CONCEPTS & ARCHITECTURE**

A **database** is a collection of related data. For example, consider the names, telephone

numbers, and addresses of the people you know. A database represents some aspect of the real world, sometimes called the miniworld.

A database management system (DBMS)is a collection of programs that enables

users to create and maintain a database. The DBMS is a *general-purpose software system*

that facilitates the processes of *defining, constructing, manipulating,* and *sharing*

databases among various users and applications.

The three-schema architecture can be used to further explain the concept of data independence, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. We can define two types of data independence:Logical data independence, Physical data independence.

the variety of users supported by a DBMS. The DBMS must provide appropriate languages and interfaces for each category of users. In this section we discuss the types of languages and interfaces provided by a DBMS and the user categories targeted by each interface.

A DBMS is a complex software system. In this section we discuss the types of software components that constitute a DBMS and the types of computer system software with which the DBMS interacts.

We distinguished the schema, or description of a database, from the database itself.

The schema does not change very often, whereas the database state changes every time data is inserted, deleted, or modified.

 Then we described the three-schema DBMS architecture, which allows three schema levels:

■ An internal schema describes the physical storage structure of the database.

■ A conceptual schema is a high-level description of the whole database.

■ External schemas describe the views of different user groups.

**Topic 2: DATA MODELING**

We defined a data model and we distinguished three main categories:

■ High-level or conceptual data models (based on entities and relationships)

■ Low-level or physical data models

■ Representational or implementation.

We present the modeling concepts of the **Entity-Relationship** (**ER**) **model**, which is a popular high-level conceptual data model. This model and its variations are frequently used for the conceptual design of database applications, and many database design tools

employ its concepts Inheritance.

**Specialization** is the process of defining a *set of subclasses* of an entity type; this entity type is called the **superclass** of the specialization. We can think of a *reverse process* of abstraction in which we suppress the differences among several entity types, identify their common features, and **generalize** them into a single **superclass** of which the original entity types are special **subclasses**.

ER Diagram Representation Now we shall learn how ER Model is represented by means of ER diagram. Every object like entity, attributes of an entity, relationship set, and attributes of relationship set can be represented by tools of ER diagram. Entity Entities are represented by means of rectangles. Rectangles are named with the entity set they represent Attributes Attributes are properties of entities. Attributes are represented by means of eclipses. Every eclipse represents one attribute and is directly connected to its entity (rectangle).

Generalization : the process of generalizing entities, where the generalized entities contain the properties of all the generalized entities is called Generalization. In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics. For an example, pigeon, house sparrow, crow and dove all can be generalized as Birds.

Specialization: Specialization is a process, which is opposite to generalization, as mentioned above. In specialization, a group of entities is divided into sub-groups based on their characteristics. Take a group Person for example. A person has name, date of birth, gender etc. These properties are common in all persons, human beings. But in a company, a person can be identified as employee, employer, customer or vendor based on what role do they play in company.

Inheritance We use all above features of ER-Model, in order to create classes of objects in object oriented programming. This makes it easier for the programmer to concentrate on what she is programming. Details of entities are generally hidden from the user, this process known as abstraction. One of the important features of Generalization and Specialization, is inheritance, that is, the attributes of higher-level entities are inherited by the lower level entities.

**Topic 3: RELATIONAL DATA MODEL**

Relation Data Model Relational data model is the primary data model, which is used widely around the world for data storage and processing. This model is simple and have all the properties and capabilities required to process data with storage efficiency. Concepts Tables: In relation data model, relations are saved in the format of Tables. This format stores the relation among entities. A table has rows and columns, where rows represent records and columns represents the attributes. Tuple: A single row of a table, which contains a single record for that relation is called a tuple.

 Relation instance: A finite set of tuples in the relational database system represents relation instance. Relation instances do not have duplicate tuples. Relation schema: This describes the relation name (table name), attributes and their names. Relation key: Each row has one or more attributes which can identify the row in the relation (table) uniquely, is called the relation key. Attribute domain: Every attribute has some pre-defined value scope, known as attribute domain.

Constraints Every relation has some conditions that must hold for it to be a valid relation. These conditions are called Relational Integrity Constraints. There are three main integrity constraints. • Key Constraints • Domain constraints • Referential integrity constraints Key Constraints: There must be at least one minimal subset of attributes in the relation, which can identify a tuple uniquely. This minimal subset of attributes is called key for that relation. If there are more than one such minimal subsets, these are called candidate keys. Key constraints forces that: • in a relation with a key attribute, no two tuples can have identical value for key attributes. • key attribute can not have NULL values.

Key constrains are also referred to as Entity Constraints. Domain constraints Attributes have specific values in real-world scenario. For example, age can only be positive integer. The same constraints has been tried to employ on the attributes of a relation. Every attribute is bound to have a specific range of values. For example, age can not be less than zero and telephone number can not be a outside 0-9. Referential integrity constraints This integrity constraints works on the concept of Foreign Key. A key attribute of a relation can be referred in other relation, where it is called foreign key. Referential integrity constraint states that if a relation refers to an key attribute of a different or same relation, that key element must exists.

Relational database systems are expected to be equipped by a query language that can assist its user to query the database instances. This way its user empowers itself and can populate the results as required. There are two kinds of query languages, relational algebra and relational calculus.

Relational algebra Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either unary or binary. They accept relations as their input and yields relations as their output. Relational algebra is performed recursively on a relation and intermediate results are also considered relations Fundamental operations of Relational algebra: • Select • Project • Union • Set different • Cartesian product • Rename

**Topic 4: ER AND EER TO RELATIONAL MAPPING**

ER Model when conceptualized into diagrams gives a good overview of entity-relationship, which is easier to understand. ER diagrams can be mapped to Relational schema that is, it is possible to create relational schema using ER diagram. Though we cannot import all the ER constraints into Relational model but an approximate schema can be generated. There are more than one processes and algorithms available to convert ER Diagrams into Relational Schema. Some of them are automated and some of them are manual process. We may focus here on the mapping diagram contents to relational basics. ER Diagrams mainly comprised of: • Entity and its attributes • Relationship, which is association among entities.

The EER model includes *all the modeling concepts of the ER model.* it includes the concepts of subclass and superclass and the related concepts of specialization and generalization

**Topic 5:** **DATABASE DESIGN**

Normalization If a database design is not perfect it may contain anomalies, which are like a bad dream for database itself. Managing a database with anomalies is next to impossible. • Update anomalies: if data items are scattered and are not linked to each other properly, then there may be instances when we try to update one data item that has copies of it scattered at several places, few instances of it get updated properly while few are left with their old values. This leaves database in an inconsistent state. • Deletion anomalies: we tried to delete a record, but parts of it left undeleted because of unawareness, the data is also saved somewhere else. • Insert anomalies: we tried to insert data in a record that does not exist at all.

Normalization is a method to remove all these anomalies and bring database to consistent state and free from any kinds of anomalies.

First Normal Form: This is defined in the definition of relations (tables) itself. This rule defines that all the attributes in a relation must have atomic domains. Values in atomic domain are indivisible units.

Second normal form says, that every non-prime attribute should be fully functionally dependent on prime key attribute. That is, if X → A holds, then there should not be any proper subset Y of X, for that Y → A also holds.

Third Normal Form: For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy: • No non-prime attribute is transitively dependent on prime key attribute • For any non-trivial functional dependency, X → A, then either • X is a superkey or, • A is prime attribute.

Boyce-Codd Normal Form: BCNF is an extension of Third Normal Form in strict way. BCNF states that • For any non-trivial functional dependency, X → A, then X must be a super-key.

**Topic 6: DATABASE SECURITY & AUTHORIZATION**

Database Security deals with all various aspects of protecting the database content, its owners, and its users. It ranges from protection from intentional unauthorized database uses to unintentional database accesses by unauthorized entities (e.g., a person or a computer program).

Authorization permits only certain users to access, process, or alter data; it also creates limitations on user access or actions.

The limitations placed on (or removed from) users can apply to objects such as schemas, entire tables, or table rows.

A user privilege is the right to run a particular type of SQL statement, or the right to access an object that belongs to another user, run a PL/SQL package, and so on. The types of privileges are defined by Oracle Database.

Roles are created by users (usually administrators) to group together privileges or other roles. They are a way to facilitate the granting of multiple privileges or roles to users.

Privileges can fall into the following general categories:

**System privileges.** These privileges allow the grantee to perform standard administrator tasks in the database. Restrict them only to trusted users. See the following sections describe privileges,etc.

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Third Semester**

**Session 2017 – 2018**

**Paper – KU BCA 302- SOFTWARE ENGINEERING PRINCIPLES**

 **Topic 1: Introduction to Software Engineering:**

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures

Let us first understand what software engineering stands for. The term is made of two words, software and engineering.

**Software** is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product.**

**Engineering** on the other hand, is all about developing products, using well-defined, scientific principles and methods.

**Software engineering** is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

Characteristics of good software

A software product can be judged by what it offers and how well it can be used. This software must satisfy on the following grounds:

* Operational,Transitional,Maintenance

Well-engineered and crafted software is expected to have the following characteristics:

Operational

This tells us how well software works in operations. It can be measured on:

1. Budget 2.Usability 3.Efficiency4.Correctness5.FunctionalityDependability5.SecuritySafety

Transitional

This aspect is important when the software is moved from one platform to another:

1.Portability 2.Interoperability 3.Reusability

* Adaptability

Maintenance

This aspect briefs about how well a software has the capabilities to maintain itself in the ever-changing environment:

1.Modularity 2.Maintainability 3.Flexibility 4.Scalability

In short, Software engineering is a branch of computer science, which uses well-defined engineering concepts required to produce efficient, durable, scalable, in-budget and on-time software products.

A **software metric** is a standard of measure of a degree to which a [software](https://en.wikipedia.org/wiki/Software) system or process possesses some property. Even if a metric is not a measurement (metrics are functions, while measurements are the numbers obtained by the application of metrics), often the two terms are used as synonyms. Since quantitative measurements are essential in all sciences, there is a continuous effort by [computer science](https://en.wikipedia.org/wiki/Computer_science%22%20%5Co%20%22Computer%20science) practitioners and theoreticians to bring similar approaches to software development. The goal is obtaining objective, reproducible and quantifiable measurements, which may have numerous valuable applications in schedule and budget planning, cost estimation, quality assurance, testing, software debugging, software performance optimization, and optimal personnel task assignments.

**Product metrics**: It describes the characteristics of the **product** such as size, compatibility, design features, performance and quality level. **Process metrics**: It can be used to improve efficiency of an existing **process** used in software development and maintenance.

**Topic 2: Software Life Cycle Models**

The software development paradigm helps developer to select a strategy to develop the software. A software development paradigm has its own set of tools, methods and procedures, which are expressed clearly and defines software development life cycle. A few of software development paradigms or process models are defined as follows:

Waterfall Model

Waterfall model is the simplest model of software development paradigm. It says the all the phases of SDLC will function one after another in linear manner. That is, when the first phase is finished then only the second phase will start and so on.

Spiral Model

Spiral model is a combination of both, iterative model and one of the SDLC model. It can be seen as if you choose one SDLC model and combine it with cyclic process (iterative model). This model considers risk, which often goes un-noticed by most other models.

**Topic 3: Software Requirements Specification:**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

The process to gather the software requirements from client, analyze and document them is known as requirement engineering.

The goal of requirement engineering is to develop and maintain sophisticated and descriptive ‘System Requirements Specification’ document.

Software Documentation

Software documentation is an important part of software process. A well written document provides a great tool and means of information repository necessary to know about software process. Software documentation also provides information about how to use the product.

A well-maintained documentation should involve the following documents:

* **Requirement documentation** - This documentation works as key tool for software designer, developer and the test team to carry out their respective tasks. This document contains all the functional, non-functional and behavioral description of the intended software.

Source of this document can be previously stored data about the software, already running software at the client’s end, client’s interview, questionnaires and research. Generally it is stored in the form of spreadsheet or word processing document with the high-end software management team.

This documentation works as foundation for the software to be developed and is majorly used in verification and validation phases. Most test-cases are built directly from requirement documentation.

* **Software Design documentation** - These documentations contain all the necessary information, which are needed to build the software. It contains: **(a)** High-level software architecture, **(b)** Software design details, **(c)** Data flow diagrams, **(d)** Database design

These documents work as repository for developers to implement the software. Though these documents do not give any details on how to code the program, they give all necessary information that is required for coding and implementation.

* **Technical documentation** - These documentations are maintained by the developers and actual coders. These documents, as a whole, represent information about the code. While writing the code, the programmers also mention objective of the code, who wrote it, where will it be required, what it does and how it does, what other resources the code uses, etc.

The technical documentation increases the understanding between various programmers working on the same code. It enhances re-use capability of the code. It makes debugging easy and traceable.

There are various automated tools available and some comes with the programming language itself. For example java comes JavaDoc tool to generate technical documentation of code.

* **User documentation** - This documentation is different from all the above explained. All previous documentations are maintained to provide information about the software and its development process. But user documentation explains how the software product should work and how it should be used to get the desired results.

These documentations may include, software installation procedures, how-to guides, user-guides, uninstallation method and special references to get more information like license updation etc.

Software Requirements Characteristics

Gathering software requirements is the foundation of the entire software development project. Hence they must be clear, correct and well-defined.

A complete Software Requirement Specifications must be:

* Clear,Correct,Consistent,Coherent,Comprehensible,Modifiable,Verifiable,Prioritized,Unambiguous

**Structure of a Requirements Document**

Requirements have to be specified using some specification language. Though formal notations exist for specifying specific properties of the system, natural languages are now most often used for specifying requirements. When formal languages are employed, they are often used to specify particular properties orfor specific parts of the system, as part of the overall SRS.

All the requirements for a system, stated using a formal notation or natural language, have to be included in a document that is clear and concise. For this, it is necessary to properly organize the requirements document.

The general structure of an SRS is given below

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

3.6 Specific Requirements

3. Detailed Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2. Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.5Attributes

3.6 Other Requirements

**Topic 5: Software Quality Assurance:** ISO 9000 and SEI CMM and their Comparison.

The Capability Maturity Model for Software (CMM), developed by the Software Engineering Institute, and the ISO 9000 series of standards, developed by the International Standards Organization, share a common concern with quality and process management. The two are driven by similar concerns and intuitively correlated. The purpose of this report is to contrast the CMM and ISO 9001, showing both their differences and their similarities. The results of the analysis indicate that, although an ISO 9001-compliant organization would not necessarily satisfy all of the level 2 key process areas, it would satisfy most of the level 2 goals and many level 3 goals. Because there are practices in the CMM that are not addressed in ISO 9000, it is possible for a level 1 organization to receive 9001 registration; similarly, there are areas addressed by ISO 9001 that are not addressed in the CMM. A level 3 organization would have little difficulty in obtaining ISO 9001 certification, and a level 2 organization would have significant advantages in obtaining certification.

**Topic 6: Software Development Tools:**

**ROSE** = **Rational** Object Oriented **Software Engineering**. ● **Rational Rose** is a set of visual modeling tools for. development of object oriented **software**. ● Visual Modeling is the process of graphically depicting the system to be developed.

Rational Rose is an [object-oriented](http://searchsoa.techtarget.com/definition/object-oriented-programming) Unified Modeling Language ([UML](http://searchsoftwarequality.techtarget.com/definition/Unified-Modeling-Language)) software design tool intended for visual modeling and [component](http://whatis.techtarget.com/definition/component) construction of enterprise-level software applications. In much the same way a theatrical director blocks out a play, a software designer uses Rational Rose to visually create (model) the framework for an application by blocking out classes with actors (stick figures), [use case](http://searchsoftwarequality.techtarget.com/definition/use-case) elements (ovals), objects (rectangles) and messages/relationships (arrows) in a sequence diagram using drag-and-drop symbols. Rational Rose documents the diagram as it is being constructed and then generates code in the designer's choice of [C++](http://searchsqlserver.techtarget.com/definition/C), [Visual Basic](http://searchwindevelopment.techtarget.com/definition/Visual-Basic), [Java](http://searchsoa.techtarget.com/definition/Java), Oracle8, [Corba](http://searchsqlserver.techtarget.com/definition/CORBA) or Data Definition Language.

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Second Semester**

**Session 2017 – 2018**

**Honours Paper – KU BCA 201 DATA STRUCTURES & C++**

**Topic 1: INTRODUCTION TO C++**

C++ is Object –Oriented Programming. It is a high level language. It was developed by Bjarne Stroustrup in 1980 .it is super set of c. it is case sensitive language.

There are different types of tokens used in c++ like variable, constant etc

A variable is nothing but a name given to a storage area that our programs can manipulate.

Constants refer to fixed values that the program may not alter during its execution. These fixed values are also called literals. Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type.

**Topic 2: DATA STRUCTURE**

The term data structure is used to describe the way data is stored, and the term algorithm is used to describe the way data is processed. There are different types of data structure 1) linear and non linear data structure

2) homogeneous and non homogeneous data structure 3)static and non static data structure

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Fourth Semester**

**Session 2017 – 2018**

**Honours Paper–10(a) Theory: JAVA Programming**

**Topic 1: Introduction to Java**

James Gosling initiated Java language project in June 1991 for use in one of his many set-top box projects. The language, initially called ‘Oak’ after an oak tree that stood outside Gosling's office, also went by the name ‘Green’ and ended up later being renamed as Java, from a list of random words.

Sun released the first public implementation as Java 1.0 in 1995. It promised Write Once, Run Anywhere(WORA), providing no-cost run-times on popular platforms.

Java is: Object Oriented, Platform Independent, Simple, Secure, Portable, Robust, Multithreaded, Distributed etc.

**Topic 2:** **JDK Tools**

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

**Topic 3:** **Classes and objects**

Java is a true OO language and therefore the underlying structure of all Java programs is classes.

 Anything we wish to represent in Java must be encapsulated in a class that defines the “state” and “behaviour” of the basic program components known as objects.

Classes create objects and objects use methods to communicate between them. They provide a

convenient method for packaging a group of logically related data items and functions that work on them.

**Topic 4:** **Applets & Applications**

A Java applet is a small application which is written in Java or another programming language that compiles to Java byte code and delivered to users in the form of that byte code. The user launches the Java applet from a web page, and the applet is then executed within a Java Virtual Machine (JVM) in a process separate from the web browser itself. A Java applet can appear in a frame of the web page, a new application window, Sun's Applet Viewer, or a stand-alone tool for testing applets. Java applets were introduced in the first version of the Java language, which was released in 1995.Java applets are usually written in Java.

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Part -III**

**Session 2016 – 2017**

**Honours Paper–10(a) Theory: JAVA Programming**

**Topic 1: Introduction to Java**

James Gosling initiated Java language project in June 1991 for use in one of his many set-top box projects. The language, initially called ‘Oak’ after an oak tree that stood outside Gosling's office, also went by the name ‘Green’ and ended up later being renamed as Java, from a list of random words.

Sun released the first public implementation as Java 1.0 in 1995. It promised Write Once, Run Anywhere(WORA), providing no-cost run-times on popular platforms.

Java is: Object Oriented, Platform Independent, Simple, Secure, Portable, Robust, Multithreaded, Distributed etc.

**Topic 2:** **JDK Tools**

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

**Topic 3:** **Classes and objects**

Java is a true OO language and therefore the underlying structure of all Java programs is classes.

 Anything we wish to represent in Java must be encapsulated in a class that defines the “state” and “behaviour” of the basic program components known as objects.

Classes create objects and objects use methods to communicate between them. They provide a

convenient method for packaging a group of logically related data items and functions that work on them.

**Topic 4:** **Applets & Applications**

A Java applet is a small application which is written in Java or another programming language that compiles to Java byte code and delivered to users in the form of that byte code. The user launches the Java applet from a web page, and the applet is then executed within a Java Virtual Machine (JVM) in a process separate from the web browser itself. A Java applet can appear in a frame of the web page, a new application window, Sun's Applet Viewer, or a stand-alone tool for testing applets. Java applets were introduced in the first version of the Java language, which was released in 1995.Java applets are usually written in Java.

All Java programs can be classified as Applications and Applets. The striking differences are that applications contain main() method where as applets do not. One more is, applications can be executed at DOS prompt and applets in a browser. We can say, an applet is an **Internet application**.

**Topic 5:** **Introduction to threads**

A thread, in the context of Java, is the path followed when executing a program. All Java programs have at least one thread, known as the main thread, which is created by the JVM at the program's start, when the main() method is invoked with the main thread.

 threads share the process's resources, but are able to execute independently. The threaded programming model provides developers with a useful abstraction of concurrent execution. Multithreading can also be applied to one process to enable parallel execution on a multiprocessing system.

**Topic 6:** **Packages**

 java package is a group of similar types of classes, interfaces and sub-packages. Package in javacan be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java**,** lang, awt, javax, swing, net, io, util, sql etc.

**Department of Computer Applications**

**Gist of Lectures as per Syllabus of Fourth Semester**

**Session 2017 – 2018**

**Honours Paper–KU BCA 401 Theory: DATA COMMUNICATION AND COMPUTER NETWORK**

Topic 1:Data Transmission Basic Concepts and Terminology

A computer network or data network is a [digital](https://en.wikipedia.org/wiki/Digital_signal) [telecommunications network](https://en.wikipedia.org/wiki/Telecommunications_network) which allows [nodes](https://en.wikipedia.org/wiki/Node_%28networking%29) to share resources. In computer networks, networked [computing devices](https://en.wikipedia.org/wiki/Computing_device) [exchange data](https://en.wikipedia.org/wiki/Data_transmission) with each other using a [data link](https://en.wikipedia.org/wiki/Data_link). The connections between nodes are established using either [cable media](https://en.wikipedia.org/wiki/Networking_cables) or [wireless media](https://en.wikipedia.org/wiki/Wireless_network).

After invention of computer there was a need of sharing data among the computer systems so that necessity helps in developing network through which not only a city and a country connected but also whole world is connected.Network is broadly categorized into two.on the basis of transmission technology it is divided into two point to point and Broadcast networks and on the basis of geographical coverage it is divided into three LAN,MAN and WAN.

The Transmission takes place via wired medium is known as guided media

There are basically three guided medium: twisted pair –signals in electrical form, co-axial cable-signals in electrical form and optical fiber-signals in optical form.

In wired network, these are the mediums through which data is transmitted.

Topic 2: Computer Network

A network is a set of devices (often referred to as *nodes)* connected by communication

links. A node can be a computer, printer, or any other device capable of sending and/or

receiving data generated by other nodes on the network.

The term *physical topology* refers to the way in which a network is laid out physically



The Transmission takes place via wired medium is known as guided media

There are basically three guided medium: twisted pair –signals in electrical form, co-axial cable-signals in electrical form and optical fiber-signals in optical form.

In wired network, these are the mediums through which data is transmitted.

*Seven layers of the OSI model*

71 Application,61 Presentation,51 Session,41 Transport,31 Network,21 Data link,1 I Physical

Topic 3: Network Security

Data Encryption Standard (DES) was designed by IBM and adopted by the U.S. government as the standard encryption method for nonmilitary and non classified use.

A firewall is a device (usually a router or a computer) installed between the internal network of an organization and the rest of the Internet. It is designed to forward some packets and filter (not forward) others

Topic 4: Basic Applications:

Terminal Network (TELNET) A general purpose client-server program that allows

remote login.

TELNET is an abbreviation for *TErminaL NETwork.* It is the standard TCP/IP protocol for virtual

terminal service as proposed by the International Organization for Standards (ISO).

TELNET enables the establishment of a connection to a remote system in such a way

that the local terminal appears to be a terminal at the remote system.

File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for

copying a file from one host to another. Although transferring files from one system to

another seems simple and straightforward, some problems must be dealt with first. For

example, two systems may use different file name conventions. Two systems may have

different ways to represent text and data. Two systems may have different directory

structures. All these problems have been solved by FTP in a very simple and elegant

approach.

 SMTP:The actual mail transfer is done through message transfer agents. To send mail, a system

must have the client MTA, and to receive mail, a system must have a server MTA. The

formal protocol that defines the MTA client and server in the Internet is called the Simple

Mail Transfer Protocol (SMTP).

The Simple Network Management Protocol (SNMP) is a framework for managing

devices in an internet using the TCPIIP protocol suite. It provides a set of fundamental

operations for monitoring and maintaining an internet.

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on

the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to

FfP because it transfers files and uses the services of TCP. However, it is much simpler

than FfP because it uses only one TCP connection. There is no separate control connection;

only data are transferred between the client and the server.